

Computer Science Technical
Report CSL-TR-5-2014
April 1, 2014

Elias D. Nino-Ruiz and Adrian Sandu

*A Derivative-Free Trust Region
Framework for Variational Data
Assimilation*

Computational Science Laboratory
Computer Science Department
Virginia Polytechnic Institute and State University
Blacksburg, VA 24060
Phone: (540)-231-2193
Fax: (540)-231-6075
Email: sandu@cs.vt.edu
Web: <http://cs1.cs.vt.edu>



A Derivative-Free Trust Region Framework for Variational Data Assimilation

Elias D. Nino and Adrian Sandu
Computational Science Laboratory,
Department of Computer Science,
Virginia Polytechnic Institute and State University,
Blacksburg, VA 24060, USA
enino@vt.edu, sandu@cs.vt.edu

April 1, 2014

Abstract

This study develops a hybrid ensemble-variational approach for solving data assimilation problems. The method, called TR-4D-EnKF, is based on a trust region framework and consists of three computational steps. First an ensemble of model runs is propagated forward in time and snapshots of the state are stored. Next, a sequence of basis vectors is built and a low-dimensional representation of the data assimilation system is obtained by projecting the model state onto the space spanned by the ensemble deviations from the mean. Finally, the low-dimensional optimization problem is solved in the reduced-space using a trust region approach; the size of the trust region is updated according to the relative decrease of the reduced order surrogate cost function. The analysis state is projected back onto the full space, and the process is repeated with the current analysis serving as a new background. A heuristic approach based on the trust region size is proposed in order to adjust the background error statistics from one iteration to the next. Experimental simulations are carried out using the Lorenz and the quasi-geostrophic models. The results show that TR-4D-EnKF is an efficient computational approach, and is more accurate than the current state of the art 4D-EnKF implementations such as the POD-4D-EnKF and the Iterative Subspace Minimization methods.

Keywords: Trust Region, 4D-EnKF, Hybrid Methods

1 Introduction

Data assimilation [25] is the process of estimating the true state $\mathbf{x}_M^{\text{true}} \in \mathbb{R}^n$ of a dynamical system at the current time t_M given a history of prior evolution and noisy observations of the state at times t_k

$$\mathbf{y}_k = \mathcal{H}_k(\mathbf{x}_k^{\text{true}}) + \boldsymbol{\epsilon}_k \in \mathbb{R}^{m \times 1}, \quad 0 \leq k \leq M. \quad (1)$$

Here n is the number of components in the model state, m is the number of observed components from \mathbf{x}^{true} , $\mathcal{H}_k : \mathbb{R}^n \rightarrow \mathbb{R}^m$ is the observation operator, $\boldsymbol{\epsilon}_k \in \mathbb{R}^{m \times 1}$ is the error associated to the k -th observation time, and M is the number of observation times. Typically, observational errors are assumed to be normal distributed $\boldsymbol{\epsilon}_k \sim \mathcal{N}(\mathbf{0}_m, \mathbf{R}_k)$ where $\mathbf{0}_m$ is the m -th dimensional vector whose components are all zeros, and $\mathbf{R}_k \in \mathbb{R}^{m \times m}$ is the data error covariance matrix at the assimilation time t_k .

A dynamical model encapsulating our knowledge of the physical laws approximates the evolution of the dynamical system. The evolution of the model state \mathbf{x} is given by

$$\mathbf{x}_{k+1} = \mathcal{M}_{t_k \rightarrow t_{k+1}}(\mathbf{x}_k), \quad 0 \leq k \leq M-1, \quad (2)$$

where \mathcal{M} represents a nonlinear model solution operator (e.g., which simulates the evolution of the ocean or the atmosphere).

Two families of methods, statistical filters and variational, are widely used to solve data assimilation problems. Representative methods of those classes are the Ensemble Kalman Filter (EnKF) and the Four-Dimensional Variational Method (4D-Var), respectively. In EnKF an ensemble of model runs is propagated in time; when data is available the filtering step generates an *analysis ensemble* whose empirical mean is an estimator for \mathbf{x}^{true} . Strong constraint 4D-Var seeks an *analysis initial state* such that the corresponding forecast best fits the observations within the assimilation window. It is well-accepted that both methods face specific challenges in practical applications where $n \sim 10^9$. For instance, ensemble-based filters suffer from statistical sampling errors, while variational methods require adjoint models which are labor-intensive to develop and computationally expensive to run.

Hybrid methods have been proposed in order to combine the strengths of EnKF and 4D-Var methods. A decomposition of the background errors in components that are analyzed and components that are ignored has been used to estimate posterior covariances [3], and the theoretical similarities between the two approaches have been used to construct look-ahead assimilation techniques [26]. Other hybrid approaches are based on model reduction and/or space reduction [2, 12, 27, 28]. A discussion of model reduction techniques is given in [24]. In this paper, we focus on the reduced-space approach where a subspace of the state space is identified, the variational problem is solved in this subspace, then the analysis is projected back onto the model space. The new solution can be treated as a new background and the process is repeated. The reduced space data assimilation approaches available in the literature update the solution in the model space unconditionally. No available method provides a relation between the analysis at the current iteration and its associated error statistics (i.e., the initial background error covariance matrix is assumed to hold for all the ensembles at all iterations). This fact is important in the sampling process, as the uncertainty associated with the analysis state decreases as the iterations progress.

In this work we formulate a hybrid data assimilation algorithm in the context of derivative-free optimization. A rigorous Trust Region (TR) framework is

proposed where the TR radius in the model space is linked with the spread of the ensemble members and with the quality of the solutions found in the reduced-space. The new method is named TR-4D-EnKF.

The remainder of the paper is organized as follows. Section 2 reviews the current state of the art ensemble-based approaches to data assimilation. Section 3 develops the novel derivative free TR-4D-EnKF method. Numerical results using the Lorenz-96 and the quasi-geostrophic models are reported in Section 4, and conclusions are presented in Section 5.

2 Four-dimensional ensemble-based approaches to data assimilation

EnKF [14] is one of the most widely used methods in data assimilation due to its simple formulation and ease of implementation. Normality assumptions are made on both the background and data errors [15]. The method contains two steps, the forecast and the analysis.

The *prior* (background) distribution is approximated by an ensemble of $N+1$ model state samples

$$\mathbf{X}_0 = \left[\mathbf{x}_0^{\text{b}(1)}, \mathbf{x}_0^{\text{b}(2)}, \dots, \mathbf{x}_0^{\text{b}(N+1)} \right] \in \mathbb{R}^{n \times (N+1)}, \quad (3)$$

with the empirical moments

$$\bar{\mathbf{x}}_0 = \frac{1}{N+1} \cdot \sum_{i=1}^{N+1} \mathbf{x}_0^{\text{b}(i)} \in \mathbb{R}^{n \times 1}, \quad (4a)$$

$$\mathbf{S}_0 = \frac{1}{N} \cdot \delta \mathbf{X}_0 \cdot \delta \mathbf{X}_0^T \in \mathbb{R}^{n \times n}, \quad (4b)$$

where $\mathbf{x}_0^{\text{b}(i)}$ is the i -th ensemble member and the columns of matrix

$$\delta \mathbf{X}_0 = \left[\delta \mathbf{x}_0^{(1)}, \delta \mathbf{x}_0^{(2)}, \dots, \delta \mathbf{x}_0^{(N+1)} \right] \in \mathbb{R}^{n \times (N+1)} \quad (5)$$

are given by $\delta \mathbf{x}_0^{(i)} = \mathbf{x}_0^{\text{b}(i)} - \bar{\mathbf{x}}_0$, for $1 \leq i \leq N+1$. Prior any measurement, the background state $\mathbf{x}_0^{\text{b}} \approx \bar{\mathbf{x}}_0$ provides the best estimation to $\mathbf{x}_0^{\text{true}}$.

In the forecast step the background ensemble (3) is obtained by an ensemble of model runs that propagate each model state to the current time t_k .

In the analysis step a *posterior* (analysis) ensemble is constructed by making use of the observation \mathbf{y}_k and by applying the Kalman filter to each background ensemble member:

$$\mathbf{x}_k^{\text{a}(i)} = \mathbf{x}_k^{\text{b}(i)} + \mathbf{K} \left[\mathbf{y}_k^{s(i)} + \boldsymbol{\epsilon}_k^{s(i)} - \mathbf{H}_k \cdot \mathbf{x}_k^{\text{b}(i)} \right], \quad 1 \leq i \leq N+1, \quad (6)$$

where $\mathcal{H}'_k = \mathbf{H}_k \in \mathbb{R}^{m \times n}$ is a linearized observation operator at time t_k , $\mathbf{y}_k^{s(i)} \sim \mathcal{N}(\mathbf{y}_k, \mathbf{R}_k)$ are the observations \mathbf{y}_k with added synthetic noise $\boldsymbol{\epsilon}_k^{s(i)} \sim$

$\mathcal{N}(\mathbf{0}_m, \mathbf{R}_k)$, and the Kalman gain matrix is $\mathbf{K} = \mathbf{S}_k \cdot \mathbf{H}_k^T [\mathbf{H}_k \cdot \mathbf{S}_k \cdot \mathbf{H}_k^T + \mathbf{R}_k]^{-1} \in \mathbb{R}^{n \times m}$. The ensemble members are further propagated in time

$$\mathbf{x}_{k+1}^{b(i)} := \mathcal{M}_{t_k \rightarrow t_{k+1}} \left(\mathbf{x}_k^{a(i)} \right), \quad (7)$$

to obtain the background ensemble for the forecast step. EnKF can provide flow-dependent error estimates of the background errors (with the Monte Carlo methods) [22, 23], but it does not have the ability to assimilate the observation data available at distributed times.

4D-Var considers cost functions of the form

$$\mathcal{J}(\mathbf{x}_0) = \underbrace{\frac{1}{2} \|\mathbf{x}_0 - \mathbf{x}_0^b\|_{\mathbf{B}_0^{-1}}^2}_{\mathcal{J}^b(\mathbf{x})} + \underbrace{\frac{1}{2} \sum_{k=0}^M \|\mathbf{y}_k - \mathcal{H}(\mathbf{x}_k)\|_{\mathbf{R}_k^{-1}}^2}_{\mathcal{J}^o(\mathbf{x})}, \quad (8)$$

where $\mathcal{J}^b(\mathbf{x})$ and $\mathcal{J}^o(\mathbf{x})$ are known as the background and observation cost functions, respectively. The cost function (8) is the negative logarithms of the a posteriori probability density when all the data and background errors are normally distributed. The maximum likelihood estimate of the initial state is then obtained by minimizing the cost function, i.e., the analysis step is computed by solving the optimization problem

$$\mathbf{x}_0^a = \arg \min_{\mathbf{x}_0} \mathcal{J}(\mathbf{x}_0) \quad \text{subject to (2)}. \quad (9)$$

The formulation of (8) allows 4D-Var to assimilate data which appears at different observation times.

The computation of the gradient (8) with respect to the control variable $\mathbf{x}_0 \in \mathbb{R}^{n \times 1}$ requires one forward and one adjoint model integration. The construction of an adjoint model for real, large forecast models is an extremely labor-intensive process. In order to avoid the implementation of adjoint models four dimensional ensemble Kalman filter methods (4D-EnKF) [32] have been recently proposed. They naturally propagate flow dependent background covariance matrices via ensembles [17, 21, 27, 4]. Numerical experiments show robust performance with a small number of ensemble members [31, 30]. Moreover, the solution (9) can be treated as the new background state in (8), which provides a better solution [1].

4D-EnKF based methods are defined as follows. The initial ensemble (3) is propagated in time and $M + 1$ snapshots of each background ensemble member state at time moments t_0, t_1, \dots, t_M along the trajectory are stored

$$\mathbf{X}^s = \begin{bmatrix} \mathbf{x}_0^{b(1)} & \mathbf{x}_0^{b(2)} & \dots & \mathbf{x}_0^{b(N+1)} \\ \mathbf{x}_1^{b(1)} & \mathbf{x}_1^{b(2)} & \dots & \mathbf{x}_1^{b(N+1)} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{x}_M^{b(1)} & \mathbf{x}_M^{b(2)} & \dots & \mathbf{x}_M^{b(N+1)} \end{bmatrix} \in \mathbb{R}^{(n \cdot (M+1)) \times (N+1)}. \quad (10)$$

Each entry of the background ensemble matrix \mathbf{X}^s is an n -dimensional vector $\mathbf{x}_k^{b(i)}$ which represents the state of ensemble member i at time t_k . The i -th column of \mathbf{X}^s contains all the snapshots of the i -th ensemble member, and the k -th row of blocks corresponds to all ensemble member states at t_k .

Consider now a trajectory of the model. The state \mathbf{x}_k at t_k is approximated by a linear combination of the anomalies (deviations from the mean)

$$\mathbf{x}_k = \bar{\mathbf{x}}_k + \sum_{i=1}^N \alpha_i \cdot \underbrace{(\mathbf{x}_k^{b(i)} - \bar{\mathbf{x}}_k)}_{\boldsymbol{\psi}_k^{(i)}} = \bar{\mathbf{x}}_k + \boldsymbol{\Psi}_k \cdot \boldsymbol{\alpha}, \quad (11)$$

where

$$\bar{\mathbf{x}}_k = \frac{1}{N+1} \cdot \sum_{i=1}^{N+1} \mathbf{x}_k^{b(i)} \in \mathbb{R}^{n \times 1}, \quad (12)$$

$$\boldsymbol{\Psi}_k = [\boldsymbol{\psi}_k^{(1)}, \boldsymbol{\psi}_k^{(2)}, \dots, \boldsymbol{\psi}_k^{(N)}] \in \mathbb{R}^{n \times N}, \quad (13)$$

and the time-independent weight vector

$$\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \dots, \alpha_N]^T \in \mathbb{R}^{N \times 1},$$

contains the coordinates of \mathbf{x}_k in the ensemble space.

By replacing (11) in (8) and linearizing the observation operator $\mathcal{H}_k \approx \mathbf{H}_k$, the 4D-Var cost function (8) can be written in the ensemble space as follows:

$$\mathcal{J}_{\text{ens}}(\boldsymbol{\alpha}) = \frac{1}{2} \|\mathbf{d}^b - \boldsymbol{\Psi}_0 \cdot \boldsymbol{\alpha}\|_{\mathbf{B}_0^{-1}}^2 + \frac{1}{2} \sum_{k=0}^M \|\mathbf{d}_k^o - \mathbf{Q}_k \cdot \boldsymbol{\alpha}\|_{\mathbf{R}_k^{-1}}^2 \quad (14)$$

where $\mathbf{d}^b = \mathbf{x}_0^b - \bar{\mathbf{x}}_0 \in \mathbb{R}^{n \times 1}$ and $\mathbf{d}_k^o = \mathbf{y}_k - \mathbf{H}_k \cdot \bar{\mathbf{x}}_k \in \mathbb{R}^{m \times 1}$ are the innovation vectors on the background and observations, respectively, and $\mathbf{Q}_k = \mathbf{H}_k \cdot \boldsymbol{\Psi}_k \in \mathbb{R}^{m \times N}$.

The optimal solution in the ensemble space

$$\boldsymbol{\alpha}^* = \arg \min_{\boldsymbol{\alpha}} \mathcal{J}_{\text{ens}}(\boldsymbol{\alpha}) \in \mathbb{R}^{N \times 1}, \quad (15)$$

provides an approximation of the analysis trajectory started from (9) through the relation

$$\mathbf{x}_k^a = \mathbf{x}_k^b + \boldsymbol{\Psi}_k \cdot \boldsymbol{\alpha}^* \in \mathbb{R}^{n \times 1}. \quad (16)$$

The derivatives of (14) are

$$\begin{aligned} \nabla_{\boldsymbol{\alpha}} \mathcal{J}_{\text{ens}}(\boldsymbol{\alpha}) &= \left[\boldsymbol{\Psi}_0^T \cdot \mathbf{B}_0^{-1} \cdot \boldsymbol{\Psi}_0 + \sum_{k=0}^M \mathbf{Q}_k^T \cdot \mathbf{R}_k \cdot \mathbf{Q}_k \right] \cdot \boldsymbol{\alpha} \\ &- \left[\boldsymbol{\Psi}_0^T \cdot \mathbf{B}_0^{-1} \cdot \mathbf{d}^b + \sum_{k=0}^M \mathbf{Q}_k^T \cdot \mathbf{R}_k^{-1} \cdot \mathbf{d}_k \right] \in \mathbb{R}^{N \times 1}, \end{aligned} \quad (17a)$$

$$\nabla_{\boldsymbol{\alpha}, \boldsymbol{\alpha}}^2 \mathcal{J}_{\text{ens}}(\boldsymbol{\alpha}) = \boldsymbol{\Psi}_0^T \cdot \mathbf{B}_0^{-1} \cdot \boldsymbol{\Psi}_0 + \sum_{k=0}^M \mathbf{Q}_k^T \cdot \mathbf{R}_k \cdot \mathbf{Q}_k \in \mathbb{R}^{N \times N}, \quad (17b)$$

and the solution of the quadratic minimization problem (15) is

$$\boldsymbol{\alpha}^* = \nabla_{\boldsymbol{\alpha}, \boldsymbol{\alpha}}^2 \mathcal{J}_{\text{ens}}(\boldsymbol{\alpha})^{-1} \cdot \left[\boldsymbol{\Psi}_0^T \cdot \mathbf{B}_0^{-1} \cdot \mathbf{d}^b + \sum_{k=0}^M \mathbf{Q}_k^T \cdot \mathbf{R}_k^{-1} \cdot \mathbf{d}_k \right]. \quad (18)$$

Since \mathbf{x}_k^a in (16) represents an approximated solution rather than an exact solution, the initial analysis \mathbf{x}_0^a is only recovered and propagated in time in order to obtain an approximation of the optimal trajectory of (8).

Equivalent bases for the range of $\boldsymbol{\Psi}_k$ can be utilized to formulate the subspace approximation (11). For instance, the proper orthogonal decomposition (POD) [29] is widely used to obtain a basis that captures most of the variance of the snapshot (10). Consider the matrix of snapshots deviations

$$\delta \mathbf{X}^s = \frac{1}{\sqrt{N}} [\boldsymbol{\Psi}_0^T, \boldsymbol{\Psi}_1^T, \dots, \boldsymbol{\Psi}_M^T]^T \in \mathbb{R}^{(n \cdot (M+1)) \times N},$$

and its singular value decomposition (SVD)

$$\delta \mathbf{X}^s = \mathbf{U} \cdot \boldsymbol{\Sigma} \cdot \mathbf{V} \in \mathbb{R}^{(n \cdot (M+1)) \times N}, \quad (19)$$

where $\mathbf{U} \in \mathbb{R}^{(n \cdot (M+1)) \times (n \cdot (M+1))}$ and $\mathbf{V} \in \mathbb{R}^{N \times N}$ are the right and left singular vectors, respectively, and $\boldsymbol{\Sigma} = \text{diag}\{\sigma_1, \sigma_2, \dots, \sigma_N\} \in \mathbb{R}^{(n \cdot (M+1)) \times N}$ is a diagonal matrix whose diagonal entries are the singular values with $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_N$. Since

$$\delta \mathbf{X}^{sT} \cdot \delta \mathbf{X}^s = \mathbf{V} \cdot \boldsymbol{\Sigma}^2 \cdot \mathbf{V}^T \in \mathbb{R}^{N \times N},$$

the POD basis vectors can be computed as

$$\boldsymbol{\Phi}_k = \boldsymbol{\Psi}_k \cdot \mathbf{V} \cdot \boldsymbol{\Sigma}^{-1/2} \in \mathbb{R}^{n \times N}, \quad (20)$$

and therefore, equivalent to (15), \mathbf{x}_k can be expressed as follows:

$$\mathbf{x}_k = \bar{\mathbf{x}}_k + \sum_{i=1}^r \beta_i \cdot \left(\frac{\boldsymbol{\Psi}_k \cdot \mathbf{v}_i}{\sqrt{\sigma_i}} \right) = \bar{\mathbf{x}}_k + \boldsymbol{\Phi}_k^r \cdot \boldsymbol{\beta}, \quad (21)$$

where we have chosen the columns of $\boldsymbol{\Sigma}$ to be orthonormal, $\boldsymbol{\Phi}_k^r$ holds the first r basis vectors, $\boldsymbol{\beta} = [\beta_1, \beta_2, \dots, \beta_r]^T \in \mathbb{R}^{r \times 1}$ is the vector of weights to be determined, and r can be computed as follows

$$r = \arg \min_p \left\{ p, I(p) : \frac{\sum_{i=1}^p \sigma_i}{\sum_{i=1}^N \sigma_i} > \gamma : \gamma \in (0, 1) \right\}. \quad (22)$$

Note that, the parameter γ provides how much variance (sometimes called kinetic energy) we want to retain in the POD bases, commonly the values of γ ranges in (0.9, 0.95). It is well known that POD bases are the most efficient among all possible linear combinations in the sense, for a given number r of

basis vectors, POD decomposition captures the most possible variance [18, 19]. In addition, POD bases reduce the equation (14) to

$$\mathcal{J}_{\text{ens}}^{\text{POD}}(\boldsymbol{\beta}) = \frac{1}{2} \cdot N \cdot \|\boldsymbol{\beta}\|^2 + \frac{1}{2} \cdot \sum_{k=0}^M \|\mathbf{d}_k - \mathbf{Z}_k \cdot \boldsymbol{\beta}\|_{\mathbf{R}_k^{-1}}^2, \quad (23)$$

whose first and second derivatives are

$$\begin{aligned} \nabla_{\boldsymbol{\beta}} \mathcal{J}_{\text{ens}}^{\text{POD}}(\boldsymbol{\beta}) &= \left[N \cdot \mathbf{I}_{r \times r} + \sum_{k=0}^M \mathbf{Z}_k^T \cdot \mathbf{R}_k^{-1} \cdot \mathbf{Z}_k \right] \cdot \boldsymbol{\beta} \\ &\quad - \sum_{k=0}^M \mathbf{Z}_k^T \cdot \mathbf{R}_k^{-1} \cdot \mathbf{d}_k \in \mathbb{R}^{r \times 1}, \end{aligned} \quad (24a)$$

$$\nabla_{\boldsymbol{\beta}, \boldsymbol{\beta}}^2 \mathcal{J}_{\text{ens}}^{\text{POD}}(\boldsymbol{\beta}) = N \cdot \mathbf{I}_{r \times r} + \sum_{k=0}^M \mathbf{Z}_k^T \cdot \mathbf{R}_k^{-1} \cdot \mathbf{Z}_k \in \mathbb{R}^{r \times r}, \quad (24b)$$

where $\mathbf{Z}_k = \mathbf{H}_k \cdot \boldsymbol{\Phi}_k$ and $\mathbf{I}_{r \times r}$ is the identity matrix of dimension $r \times r$. Thus, an equivalent problem to (15) is

$$\boldsymbol{\beta}^* = \arg \min_{\boldsymbol{\beta}} \mathcal{J}_{\text{ens}}^{\text{POD}}(\boldsymbol{\beta}) \in \mathbb{R}^{r \times 1}, \quad (25)$$

whose solution reads:

$$\boldsymbol{\beta}^* = \nabla_{\boldsymbol{\beta}, \boldsymbol{\beta}}^2 \mathcal{J}_{\text{ens}}^{\text{POD}}(\boldsymbol{\beta})^{-1} \cdot \left[\sum_{k=0}^M \mathbf{Z}_k^{(k)T} \cdot \mathbf{R}_k^{-1} \cdot \mathbf{d}_k \right]. \quad (26)$$

Data assimilation methods that make use of the POD basis (such as, for example, POD-4D-EnKF [29]) are defined as follows:

1. **Ensemble generation.** The ensemble (3) is built centered at \mathbf{x}_0^b with covariance matrix \mathbf{B}_0 . The ensemble members are propagated and $M + 1$ snapshots of each member are saved.
2. **Basis computation.** The POD basis (20) are computed and r vectors are selected according to (22).
3. **Compute reduced-space solution.** Compute the optimal weights solution (26).
4. **Compute full-space initial condition.** Let $\mathbf{x}_0^a = \mathbf{x}_0^b + \boldsymbol{\Phi}_0^r \cdot \boldsymbol{\beta}^*$.
5. **Propagate analysis.** $\mathbf{x}_k^b = \mathcal{M}_{t_{k-1} \rightarrow t_k}(\mathbf{x}_{k-1}^a)$ for $1 \leq k \leq M$.

According to Tian [29] the POD bases capture not only the spatial structure of the state but also its temporal evolution.

The optimal solution of the POD-4D-EnKF provides an approximation of the analysis (9). The process can be continued in an iterative fashion in order to

improve the analysis; the solution of one iteration becomes the new background state for the next iteration. The idea of using a sequence of minimizations of the surrogates (14) or (23) in order to approach the minimum of (8) has been explored in the derivative-free optimization literature [8].

A rigorous implementation has been recently proposed by Gratton et al. [16]. The method is called *Iterative Subspace Minimization* (ISM) and solves iteratively the problem (8) via the projection of the full space onto the space spanned by the POD bases. The ISM method is defined as follows:

1. **Initialization.** Let $\mathbf{x}_0^{[0]} \leftarrow \bar{\mathbf{x}}_0$ (the initial background) and $j \leftarrow 0$.
2. **Ensemble generation.** The ensemble (3) is built centered at $\mathbf{x}_0^{[j]}$ with covariance matrix \mathbf{B}_0 . The ensemble members are propagated and $M + 1$ snapshots are saved.
3. **Basis computation.** The POD basis (20) are computed and r vectors are selected according to (22).
4. **Subproblem solution.** The optimization problem (25) is partially solved making use of the *Coordinate Search Method (CSM)* [11, 10], from which we obtain β^* .
5. **Solution update.** Set $\mathbf{x}_0^{[j+1]} \leftarrow \mathbf{x}_0^{[j]} + \Phi_0^r \cdot \beta^*$, $j \leftarrow j + 1$, and go to Step 2.

The ISM method solves the optimization subproblem (23) via the CSM approach which does not make use of derivative information, and therefore no optimality conditions are checked. Other methods can be used at this step. For instance, one can employ the analytical solution (25), which guarantees to obtain the local minimizer of each subproblem and reduce the total number of outer iterations and function evaluations.

The Trust Region (TR) framework can be employed in order to exploit the information brought by the derivatives of the ensemble cost functions (14) and (23) and to provide descent directions. One of the most attractive features of TR methods is that they are provably globally convergent under general assumptions [5, 6, 7]. A general overview of the TR approach is presented in the appendix A. To the best of our knowledge TR methods have not been used yet in the context of ensemble-variational data assimilation. This work develops a TR-based approach which performs a sequence of optimizations in ensemble spaces. The ensemble based partial solutions are linked to the full space solutions at each iteration. The background error statistics of the estimates obtained at each iteration are linked to the TR radius size and the spread of the underlying ensemble. The new method enjoys all these properties and is presented in the next section.

3 The TR-4D-EnKF method

In this section we develop a Trust Region 4D-EnKF (TR-4D-EnKF) approach to data assimilation. The method uses two nested loops. Outer iterations are related to forming and running an ensemble of full-size models and generating a basis. Inner iterations are related to computing search directions in the low dimensional space and minimizing the reduced cost function (8). We start with a general overview of the method and then present the computational algorithm in detail.

The initial solution in the model space is given by the initial approximation of the background $\mathbf{x}_0^{[\text{current}]} = \mathbf{x}_0^b$. For simplicity of notation we avoid the use of indices that denote outer iteration numbers and refer to the current and next iterations only. The initial ensemble (46) is drawn from $\mathcal{N}(\mathbf{x}_0^b, \mathbf{B}_0)$. The ensemble members are propagated in time and $M + 1$ snapshots are stored as we have discussed previously. We initialize the vector of weights to $\boldsymbol{\alpha} = \mathbf{0}_N$.

In order to solve the numerical optimization problem (9) we build a quadratic model for the cost function (8) optimization process. The standard approach makes use of the full space gradient, and possibly Hessian, of (8). We seek to avoid the implementation of a full adjoint model to compute exact derivatives. The idea is to approximate the derivatives of $\mathcal{J}(\mathbf{x})$ by the ensemble space derivatives (17a) and Hessian (17b). The resulting quadratic model is:

$$\begin{aligned}
\mathcal{Q}(\mathbf{s}_\alpha) &= \mathcal{J}_{\text{ens}}(\boldsymbol{\alpha} + \mathbf{s}_\alpha) \\
&= \frac{1}{2} \cdot \|\mathbf{d}^b - \boldsymbol{\Psi}_0 \cdot (\boldsymbol{\alpha} + \mathbf{s}_\alpha)\|_{\mathbf{B}_0^{-1}}^2 + \frac{1}{2} \sum_{k=0}^M \|\mathbf{d}_k^o - \mathbf{Q}_k \cdot (\boldsymbol{\alpha} + \mathbf{s}_\alpha)\|_{\mathbf{R}_k^{-1}}^2 \\
&= \underbrace{\frac{1}{2} \cdot \|\mathbf{d}^b - \boldsymbol{\Psi}_0 \cdot \boldsymbol{\alpha}\|_{\mathbf{B}_0^{-1}}^2 + \frac{1}{2} \sum_{k=0}^M \|\mathbf{d}_k^o - \mathbf{Q}_k \cdot \boldsymbol{\alpha}\|_{\mathbf{R}_k^{-1}}^2}_{\mathcal{J}_{\text{ens}}(\boldsymbol{\alpha})} \\
&\quad + \underbrace{\left\{ \left[\boldsymbol{\Psi}_0^T \cdot \mathbf{B}_0^{-1} \cdot \boldsymbol{\Psi}_0 + \sum_{k=0}^M \mathbf{Q}_k^T \cdot \mathbf{R}_k^{-1} \cdot \mathbf{Q}_k \right] \cdot \boldsymbol{\alpha} - \mathbf{c} \right\}^T}_{\nabla \mathcal{J}_{\text{ens}}(\boldsymbol{\alpha})^T \cdot \mathbf{s}_\alpha} \cdot \mathbf{s}_\alpha \\
&\quad + \underbrace{\frac{1}{2} \mathbf{s}_\alpha^T \left[\boldsymbol{\Psi}_0^T \cdot \mathbf{B}_0^{-1} \cdot \boldsymbol{\Psi}_0 + \sum_{k=0}^M \mathbf{Q}_k^T \cdot \mathbf{R}_k^{-1} \cdot \mathbf{Q}_k \right] \mathbf{s}_\alpha}_{\frac{1}{2} \mathbf{s}_\alpha^T \nabla^2 \mathcal{J}_{\text{ens}}(\boldsymbol{\alpha}) \mathbf{s}_\alpha},
\end{aligned}$$

where $\mathbf{c} = \boldsymbol{\Psi}_0^T \cdot \mathbf{B}_0^{-1} \cdot \mathbf{d}^b + \sum_{k=0}^M \mathbf{Q}_k^T \cdot \mathbf{R}_k^{-1} \cdot \mathbf{d}_k \in \mathbb{R}^{N \times 1}$. This can be rewritten as

$$\mathcal{Q}(\mathbf{s}_\alpha) = \mathcal{J}_{\text{ens}}(\boldsymbol{\alpha}) + \mathbf{s}_\alpha^T \nabla_{\boldsymbol{\alpha}} \mathcal{J}_{\text{ens}}(\boldsymbol{\alpha}) + \frac{1}{2} \mathbf{s}_\alpha^T \nabla_{\boldsymbol{\alpha}, \boldsymbol{\alpha}}^2 \mathcal{J}_{\text{ens}}(\boldsymbol{\alpha}) \mathbf{s}_\alpha. \quad (27)$$

The optimal step \mathbf{s}_α^* in the ensemble space is given by the solution of the

constrained optimization sub-problem

$$\mathbf{s}_\alpha^* = \arg \min_{\mathbf{s}_\alpha} \mathcal{Q}(\mathbf{s}_\alpha), \quad (28a)$$

$$\text{subject to } \|\Psi_0 \cdot (\alpha + \mathbf{s}_\alpha)\| \leq \Delta. \quad (28b)$$

The trust region constraint (58b) is formulated such as to use the trust region radius Δ from the full model space.

The solution of (28) provides the following trial point in the ensemble space

$$\alpha^{\text{trial}} = \alpha + \mathbf{s}_\alpha^*, \quad (29a)$$

which corresponds to the following state in the model space

$$\mathbf{x}_0^{\text{trial}} = \mathbf{x}_0^{\text{current}} + \underbrace{\Psi_0 \cdot (\alpha + \mathbf{s}_\alpha^*)}_{\delta \mathbf{x}^*}. \quad (29b)$$

The problem (28) is solved using Lagrangian multipliers. The first and second derivatives of the model (27) are

$$\nabla \mathcal{Q}(\mathbf{s}_\alpha) = \nabla \mathcal{J}_{\text{ens}}(\alpha) + \nabla^2 \mathcal{J}_{\text{ens}}(\alpha) \cdot \mathbf{s}_\alpha \in \mathbb{R}^{N \times 1}, \quad (30a)$$

and

$$\nabla^2 \mathcal{Q}(\mathbf{s}_\alpha) = \nabla^2 \mathcal{J}_{\text{ens}}(\alpha) \in \mathbb{R}^{N \times N}, \quad (30b)$$

respectively. The trust region constraint (28b) can be written as

$$\|\alpha + \mathbf{s}_\alpha\|_{\mathbf{P}}^2 - \Delta^2 + \varsigma^2 = 0, \quad (31)$$

where $\varsigma \in \mathbb{R}$ is a slack variable and $\mathbf{P} = \Psi_0^T \cdot \Psi_0 \in \mathbb{R}^{N \times N}$. Consider the Lagrangian

$$\mathcal{L}(\mathbf{s}_\alpha, \lambda, \varsigma) = \mathcal{Q}(\mathbf{s}_\alpha) + \lambda \left(\|\alpha + \mathbf{s}_\alpha\|_{\mathbf{P}}^2 - \Delta^2 + \varsigma^2 \right), \quad (32)$$

The constrained problem (28) becomes the unconstrained optimization problem

$$\mathbf{s}_\alpha^* = \arg \min_{\mathbf{s}_\alpha} \mathcal{L}(\mathbf{s}_\alpha, \lambda, \varsigma). \quad (33)$$

The stationarity conditions for (32) read:

$$\nabla \mathcal{L}(\mathbf{s}_\alpha, \lambda, \varsigma) = \begin{bmatrix} \mathcal{L}_{\mathbf{s}_\alpha}(\mathbf{s}_\alpha, \lambda, \varsigma) \\ \mathcal{L}_\lambda(\mathbf{s}_\alpha, \lambda, \varsigma) \\ \mathcal{L}_\varsigma(\mathbf{s}_\alpha, \lambda, \varsigma) \end{bmatrix} = 0, \quad (34)$$

where

$$\begin{aligned} \mathcal{L}_{\mathbf{s}_\alpha}(\mathbf{s}_\alpha, \lambda, \varsigma) &= \nabla \mathcal{Q}(\mathbf{s}_\alpha) + 2\lambda \mathbf{P} \cdot (\alpha + \mathbf{s}_\alpha) = \mathbf{0} \in \mathbb{R}^N, \\ \mathcal{L}_\lambda(\mathbf{s}_\alpha, \lambda, \varsigma) &= \|\alpha + \mathbf{s}_\alpha\|_{\mathbf{P}}^2 - \Delta^2 + \varsigma^2 = 0 \in \mathbb{R}, \\ \mathcal{L}_\varsigma(\mathbf{s}_\alpha, \lambda, \varsigma) &= 2 \cdot \lambda \cdot \varsigma = 0 \in \mathbb{R}. \end{aligned}$$

The system of nonlinear equations (34) can be solved via the Newton-Raphson method. The symmetric Jacobian of (34) reads

$$\begin{aligned} \nabla^2 \mathcal{L}(\mathbf{s}_\alpha, \lambda, \varsigma) &= \begin{bmatrix} \mathcal{L}_{\mathbf{s}_\alpha, \mathbf{s}_\alpha}(\mathbf{s}_\alpha, \lambda, \varsigma) & \mathcal{L}_{\mathbf{s}_\alpha, \lambda}(\mathbf{s}_\alpha, \lambda, \varsigma) & \mathcal{L}_{\mathbf{s}_\alpha, \varsigma}(\mathbf{s}_\alpha, \lambda, \varsigma) \\ * & \mathcal{L}_{\lambda, \lambda}(\mathbf{s}_\alpha, \lambda, \varsigma) & \mathcal{L}_{\lambda, \varsigma}(\mathbf{s}_\alpha, \lambda, \varsigma) \\ * & * & \mathcal{L}_{\varsigma, \varsigma}(\mathbf{s}_\alpha, \lambda, \varsigma) \end{bmatrix} \\ &\in \mathbb{R}^{(N+2) \times (N+2)} \end{aligned} \quad (35)$$

where

$$\begin{aligned} \mathcal{L}_{\mathbf{s}_\alpha, \mathbf{s}_\alpha}(\mathbf{s}_\alpha, \lambda, \varsigma) &= \nabla^2 \mathcal{Q}(\mathbf{s}_\alpha) + 2\lambda \cdot \mathbf{P} \in \mathbb{R}^{N \times N}, \\ \mathcal{L}_{\mathbf{s}_\alpha, \lambda}(\mathbf{s}_\alpha, \lambda, \varsigma) &= 2 \cdot \mathbf{P} \cdot (\boldsymbol{\alpha} + \mathbf{s}_\alpha) \in \mathbb{R}^{N \times 1}, \\ \mathcal{L}_{\mathbf{s}_\alpha, \varsigma}(\mathbf{s}_\alpha, \lambda, \varsigma) &= \mathbf{0} \in \mathbb{R}^{N \times 1}, \\ \mathcal{L}_{\lambda, \lambda}(\mathbf{s}_\alpha, \lambda, \varsigma) &= 0 \in \mathbb{R}, \\ \mathcal{L}_{\lambda, \varsigma}(\mathbf{s}_\alpha, \lambda, \varsigma) &= 2 \cdot \varsigma \in \mathbb{R}, \\ \mathcal{L}_{\varsigma, \varsigma}(\mathbf{s}_\alpha, \lambda, \varsigma) &= 2 \cdot \lambda \in \mathbb{R}, \end{aligned}$$

therefore, the solution of (32) can be obtained iteratively as follows:

$$\begin{bmatrix} \mathbf{s}_{\alpha[j+1]} \\ \lambda_{[j+1]} \\ \varsigma_{[j+1]} \end{bmatrix} = \begin{bmatrix} \mathbf{s}_{\alpha[j]} \\ \lambda_{[j]} \\ \varsigma_{[j]} \end{bmatrix} + \mathbf{p}_{[j]}^{\text{Newton}} \in \mathbb{R}^{(N+2) \times 1}, \quad (36)$$

where the Newton step is given by

$$\mathbf{p}_{[j]}^{\text{Newton}} = [\nabla^2 \mathcal{L}(\mathbf{s}_{\alpha[j]}, \lambda_{[j]}, \varsigma_{[j]})]^{-1} \cdot \nabla \mathcal{L}(\mathbf{s}_{\alpha[j]}, \lambda_{[j]}, \varsigma_{[j]}).$$

The computational cost of each iteration is bounded by $\mathcal{O}(N^3)$ and since N is small, this method does not imply any significant computational effort. Note that, when the full step is taken in the ensemble space

$$\|\Psi_0 \cdot \boldsymbol{\alpha}^{\text{trial}}\| \leq \Delta,$$

the iterative Newton method can be avoided and the exact solution (15) can be employed. The solution (15) also requires $\mathcal{O}(N^3)$ computations.

Then, $M + 1$ snapshots of the full model solution started from $\mathbf{x}_0^{\text{trial}}$ (29b) are stored. The following ratio is computed:

$$\rho = \frac{\mathcal{J}(\mathbf{x}^{\text{current}}) - \mathcal{J}(\mathbf{x}^{\text{trial}})}{\mathcal{Q}(\mathbf{0}_N) - \mathcal{Q}(\mathbf{s}_{\alpha^*})} = \frac{\mathcal{J}(\mathbf{x}^{\text{current}}) - \mathcal{J}(\mathbf{x}^{\text{trial}})}{\mathcal{J}_{\text{ens}}(\boldsymbol{\alpha}) - \mathcal{J}_{\text{ens}}(\boldsymbol{\alpha}^{\text{trial}})}. \quad (37)$$

Based on the ρ value, the next updates are made for the solution

$$(\mathbf{x}^{\text{current}}, \boldsymbol{\alpha}) := \begin{cases} (\mathbf{x}^{\text{current}}, \boldsymbol{\alpha}) & \text{for } \rho \leq \eta, \\ (\mathbf{x}^{\text{trial}}, \boldsymbol{\alpha}^{\text{trial}}) & \text{otherwise,} \end{cases} \quad (38)$$

and for the TR radius size

$$\Delta := \begin{cases} \Delta \cdot \gamma_{\text{dec}} & \text{for } \rho < \theta_1, \\ \Delta & \text{for } \theta_1 \leq \rho < \theta_2 \text{ or } \rho > 1, \\ \min(\Delta \cdot \gamma_{\text{inc}}, \Delta_{\text{max}}) & \text{for } \theta_2 \leq \rho \leq 1. \end{cases} \quad (39)$$

The new solutions in the model and ensemble space are utilized and a new optimization problem in (33) is solved. This process is repeated until a maximum number of inner iterations is reached or a full step is taken. Next, the current solution becomes the new background $\mathbf{x}_0^b := \mathbf{x}_0^{\text{current}}$, a new ensemble of full model solutions is generated, snapshots are taken, a new set of basis vectors is built, and the overall process is repeated.

The uncertainty associated with the new background is changed after the inner iterations since a partial assimilation of observations has been carried out. As an analogy, in the EnKF the spread of the ensemble members around the background is decreased after the analysis step. Consequently, before generating a new ensemble, we want to adjust the spread of the background errors. This is done according to the heuristic formula

$$\mathbf{B}_0 := \lambda_{\mathbf{B}}(\Delta) \cdot \mathbf{B}_0, \quad (40)$$

where $\lambda_{\mathbf{B}}(\Delta)$ is a function of the current TR radius size. Note that the TR radius is large when the decrease of the current (quadratic) model is a good predictor of the full model function decrease. In our context, if the dynamics of the full (nonlinear) model is well represented by the ensemble, the prediction done using the quadratic model $\mathcal{Q}(\mathbf{s}_{\alpha})$ is close to the actual reduction of the cost function $\mathcal{J}(\mathbf{x})$ and the TR radius is increased. In this case, we want the $\lambda_{\mathbf{B}}(\Delta)$ value to be small in order to decrease the uncertainty of the new ensemble around \mathbf{x}_0^b . Vice-versa, a small TR radius indicates that the current set of basis vectors does not represent well the dynamics of the model. The current assimilation step is not expected to decrease uncertainty; to keep the same uncertainty level for the next ensemble generation we need $\lambda_{\mathbf{B}}(\Delta) \approx 1$. Both cases are captured by the following heuristic function

$$\lambda_{\mathbf{B}}(\Delta) = \frac{\Delta_{\text{max}}}{\Delta_{\text{max}} + \Delta}, \quad (41)$$

which provides an inverse relation between the TR radius and the spread of the ensemble members. Other functions can be considered as well. In summary, when the TR radius is large the confidence in the current solution is increased

$$\lim_{\Delta \rightarrow \Delta_{\text{max}}} \lambda_{\mathbf{B}} = \frac{\Delta_{\text{max}}}{2 \cdot \Delta_{\text{max}}} = \frac{1}{2}.$$

On the other hand, when the TR size is small, the current level of background uncertainty remains unchanged for the new ensemble generation

$$\lim_{\Delta \rightarrow 0} \lambda_{\mathbf{B}} = \frac{\Delta_{\text{max}}}{\Delta_{\text{max}}} = 1.$$

The effects of the scaling of \mathbf{B}_0 on the new background ensemble are shown in the Figure 1 for a 2D example. The choice $\lambda_{\mathbf{B}} = 1$ keeps the uncertainty unchanged (Figure 1a), while $\lambda_{\mathbf{B}} = 1/2$ shrinks the spread by half (Figure 1b).

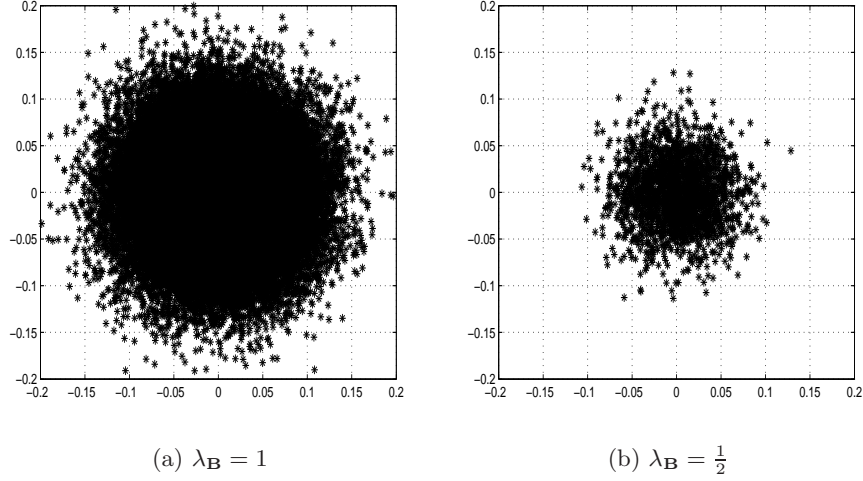


Figure 1: Impact of the scaling of \mathbf{B}_0 on the spread of the newly generated ensemble.

The outline of the TR-4D-EnKF method is shown below.

1. **Initialization.** The TR parameters are initialized. $\mathbf{x}_0^{\text{current}} := \mathbf{x}_0^{\text{b}}$ and $\boldsymbol{\alpha} = \mathbf{0}_N$.
2. **Ensemble generation.** The initial ensemble is drawn from a distribution with mean \mathbf{x}_0^{b} and background error covariance matrix \mathbf{B}_0 . M snapshots for each ensemble member are stored (10) and the basis functions (13) are computed.
3. **Inner iterations:**
 - (a) **Model construction.** Build the quadratic model (27).
 - (b) **Optimization problem.** Solve the optimization sub-problem (28) and compute $\boldsymbol{\alpha}^{\text{trial}}$ and $\mathbf{x}_0^{\text{trial}}$.
 - (c) **Ratio of prediction.** Run the full model to obtain M snapshots of the solution started from $\mathbf{x}_0^{\text{trial}}$ and compute the ratio ρ (37).
 - (d) **Solution and TR size update.** Update the solutions in the model and ensemble spaces and the TR radius according to (38) and (39), respectively. If a full step was taken or the maximum number of inner iterations is exceeded go to Step 4, otherwise go to Step 3(a).

4. **Background update.** Set $\mathbf{x}_0^b := \mathbf{x}_0^{\text{current}}$, $\boldsymbol{\alpha} := \mathbf{0}_N$, scale the covariance matrix (40), and go to Step 2.

At the end of the inner iterations the solution provides a new background for the next ensemble generation. Moreover, the background error statistics are updated according to the TR size. The outer iterations can be repeated until some stopping criterion is satisfied. The outer iteration stopping criterion can be based on the total number of iterations, on the trace of \mathbf{B}_0 , or on the trust region radius.

Now we are ready to test our implementation and compare it with other 4D-EnKF implementations discussed in Section 2.

4 Numerical experiments

In this section we study the accuracy and performance of the TR-4D-EnKF approach. First, the proposed implementation is compared with the 4D-EnKF implementations discussed in section 2: POD, SVD and ISM, using the Lorenz-96 model. Next the TR-4D-EnKF is applied to a data assimilation test problem based on a Quasi-geostrophic model. Different data error covariance matrices and linear observation operators are utilized in this test. For both cases, the methods are coded in MATLAB. The parallel toolbox of MATLAB is utilized in order to propagate the ensemble members for the different models. Realistic scenarios are assumed in all tests. The number of ensemble member is small, typically ($N \leq 80$). This is important since the model propagation is the most intensive computational part of the algorithm. Both the dimension of the vector state n and the number of observations m are much larger than the number of ensemble members.

The metrics used in the tests are the CPU time (which is reported per iteration), the cost function value (8), and the root mean square error

$$\text{RMSE} = \sqrt{\frac{1}{M} \cdot \sum_{k=0}^M (\mathbf{x}_k^{\text{true}} - \mathbf{x}_k^a)^T \cdot (\mathbf{x}_k^{\text{true}} - \mathbf{x}_k^a)}, \quad (42)$$

which provides the average of the squared root differences between the reference solution \mathbf{x}^{true} and the analysis \mathbf{x}^a over the observation times.

4.1 The Lorenz-96 model

The Lorenz 96 model is described by the following system of ordinary differential equations [9]

$$\frac{dx_i}{dt} = \begin{cases} (x_2 - x_n) \cdot x_n - x_1 + \varphi & \text{for } i = 1, \\ (x_{i+1} - x_{i-2}) \cdot x_{i-1} - x_i + \varphi & \text{for } 2 \leq i \leq n-1, \\ (x_1 - x_{n-2}) \cdot x_{n-1} - x_n + \varphi & \text{for } i = n, \end{cases} \quad (43)$$

which mimics fundamental properties of atmospheric dynamics. This model exhibits extended chaos with an external forcing value $\varphi = 8.0$, when the solution is in the form of moving waves. For this reason, the model is adequate to perform basic studies of predictability.

The experimental setting is described below.

- One time unit of the Lorenz-96 model corresponds to 1.5 days of the atmosphere.
- Snapshots are taken every 1.5 days over 150 days.
- The true (reference) initial solution $\mathbf{x}_0^{\text{true}}$ is computed numerically.
- The initial background state is built as follows

$$\mathbf{x}_0^{\text{b}} = \mathbf{x}_0^{\text{true}} + \boldsymbol{\nu}, \quad \boldsymbol{\nu} \sim \mathcal{N}(\mathbf{0}_n, \mathbf{B}_0), \quad (44)$$

where the initial background error covariance matrix \mathbf{B}_0 is given by

$$\mathbf{B}_0 = \sigma_B^2 \cdot \mathbf{I}_{n \times n}, \quad (45)$$

with $\sigma_B = 0.05$.

- The number of ensemble members is equal to $N = 40$. The initial conditions are built as follows:

$$\mathbf{x}_0^{\text{b}(i)} \sim \mathcal{N}(\mathbf{x}_0^{\text{b}}, \mathbf{B}_0), \quad 1 \leq i \leq N. \quad (46)$$

- The model propagation is performed making use of an explicit second order Runge-Kutta method. Thus, model errors have the form $\mathcal{O}(h^2)$ where h is the time-step size.
- The external force φ is set to 8.0.
- Three model resolutions n are considered: small ($n = 10^3$), medium ($n = 10^4$), and large ($n = 10^5$).
- The number of observed components equals the number of components in the vector state ($n = m$) which implies $\mathbf{H} = \mathbf{I}_{n \times n}$.
- Observations are evenly every day over 100 days (100 time units) At each observation time, the observations are built as follows

$$\mathbf{y}_k = \mathbf{H} \cdot \mathbf{x}_k^{\text{true}} + \boldsymbol{\epsilon}_k, \quad \boldsymbol{\epsilon}_k \sim \mathcal{N}(\mathbf{0}_m, \mathbf{R}), \quad 0 \leq k \leq M, \quad (47)$$

where the data error covariance matrix \mathbf{R} is given by

$$\mathbf{R} = \sigma_O^2 \cdot \mathbf{I}_{m \times m}, \quad (48)$$

with $\sigma_O = 0.01$.

- The number of outer loops for the ISM and TR-4D-EnKF is equal to 100.
- The initial parameters for the TR-4D-EnKF are $\gamma_{\text{inc}} = 1.01$, $\gamma_{\text{dec}} = 0.99$, $\Delta_{\text{max}} = 20$, $\Delta_0 = 0.1$, $\eta = 0.1$, $\theta_1 = 0.25$ and $\theta_2 = 0.75$.

n	4D-EnKF method	$\mathcal{J}(\mathbf{x})$	RMSE	CPU time/iter.
10^3	Background	2.2250×10^7	6.6623	N/A
	POD	4.2737×10^6	2.9072	21.97 s
	ISM	2.7937×10^5	0.6782	15.83 s
	TR	6.6009×10^4	0.1860	16.33 s
10^4	Background	1.8497×10^8	19.2080	N/A
	POD	5.4845×10^7	10.4290	29.17 s
	ISM	1.3560×10^7	5.1129	22.20 s
	TR	1.5994×10^6	1.4928	32.48 s
10^5	Background	1.7709×10^9	59.4288	N/A
	POD	3.3592×10^8	25.7407	127.74 s
	ISM	1.4964×10^8	17.0213	133.46 s
	TR	3.5433×10^7	7.8282	206.46 s

Table 1: Cost function values, RMSE, and CPU times for different 4D-EnKF implementations applied to the Lorenz-96 model. After 100 of outer iterations, the proposed TR implementation provides the most accurate results within a reasonable computational time per iteration.

Table 1 the accuracy and the computational effort for several 4D-EnKF implementations applied to the Lorenz-96 model are shown in. All data assimilation methods provide improvements over the background case. POD-4D-EnKF performs a single outer iteration and its analysis improves the RMSE over the background by $\sim 56\%$. The ISM method takes a more complex approach and adjusts the set of POD-basis vectors at each outer iteration. On average the ISM analyses after 100 of iterations improve the RMSE by $\sim 80\%$ over the background. The TR-4D-EnKF implementation provides a more accurate solution with the same number of outer iterations. On average, the proposed implementation improves the RMSE over the background trajectory by 92%. Figure 2 illustrates that the accuracy of the TR-4D-EnKF solutions is better than for the other implementations. Solution trajectories in state space are shown in Figures 3, 4 and 5 for $n = 10^3$, $n = 10^4$ and $n = 10^5$, respectively. The TR-4D-EnKF analysis provides the best fit to the reference solution. The TR-4D-EnKF compute solutions within a reasonable computational time; it is about as expensive as ISM for the small case and about 50% more expensive than ISM for the largest case. This is acceptable in view of the higher accuracy allowed by the method.

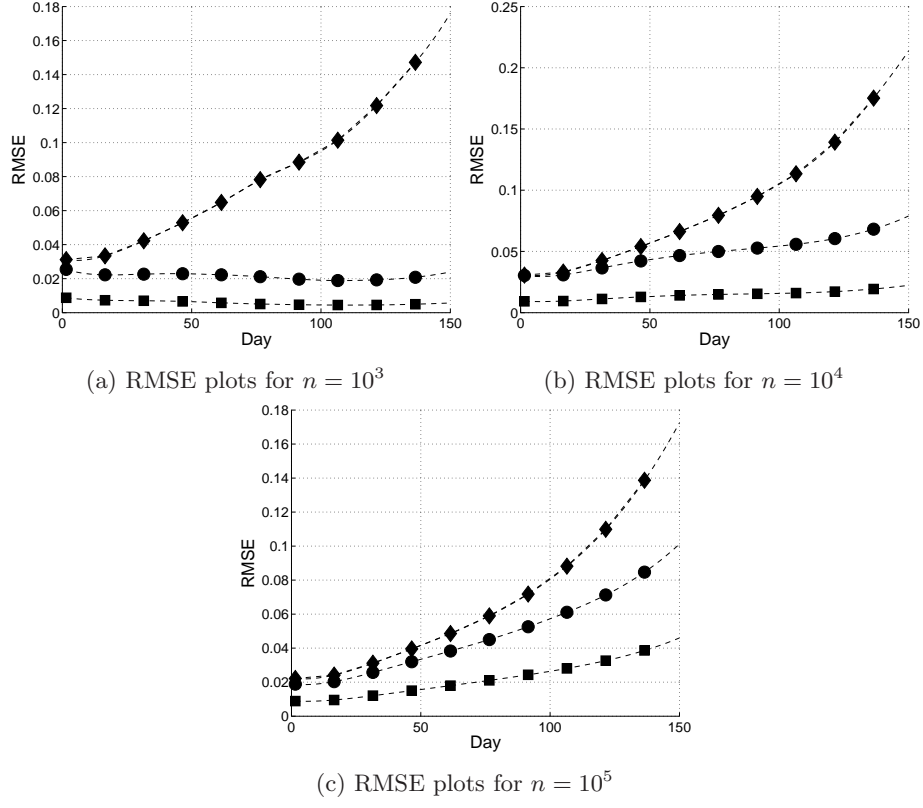


Figure 2: RMSE of the 4D-EnKF implementations POD ($-\diamond$), ISM ($-\bullet$) and TR ($-\blacksquare$) for different model resolutions (n).

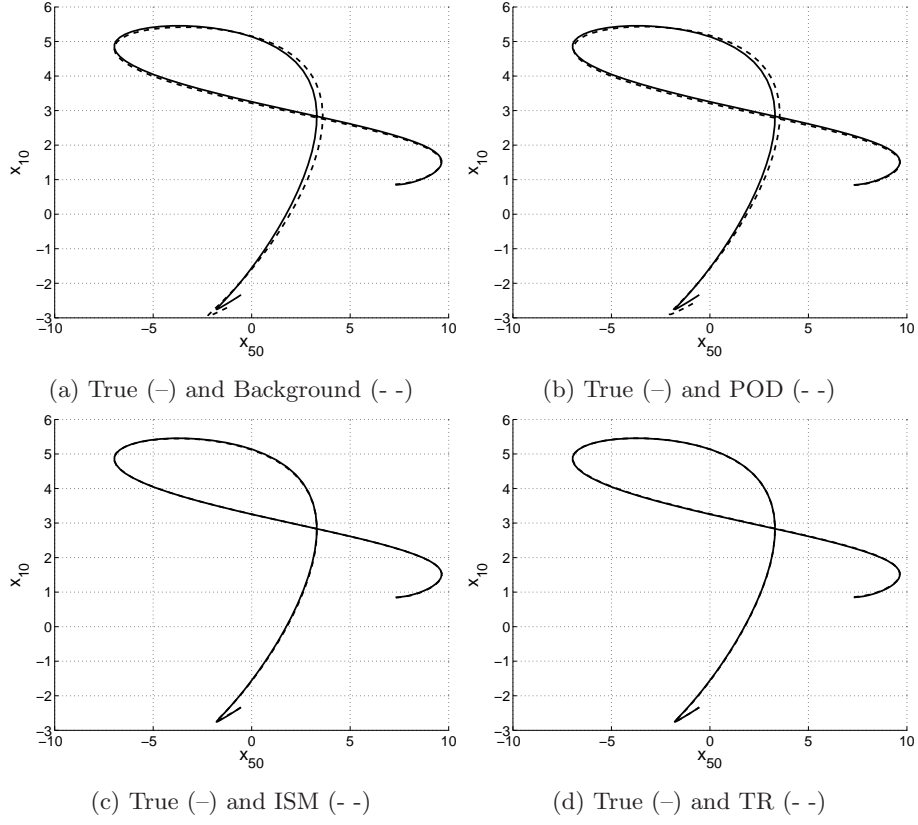


Figure 3: Snapshots of the 4D-EnKF implementations POD, SVD, ISM, and TR for the model resolution $n = 10^3$. The x_{50} and x_{10} components are plot for each method (- -) and the true solution (-).

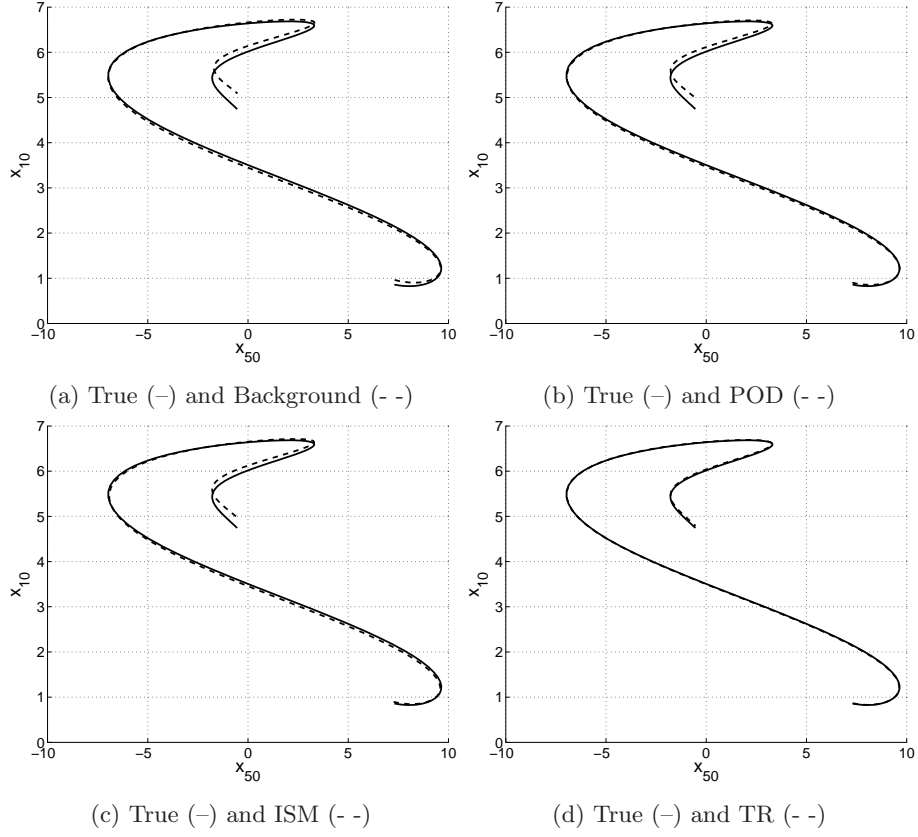


Figure 4: Snapshots of the 4D-EnKF implementations POD, SVD, ISM, and TR for the model resolution $n = 10^4$. The x_{50} and x_{10} components are plot for each method (---) and the true solution (—).

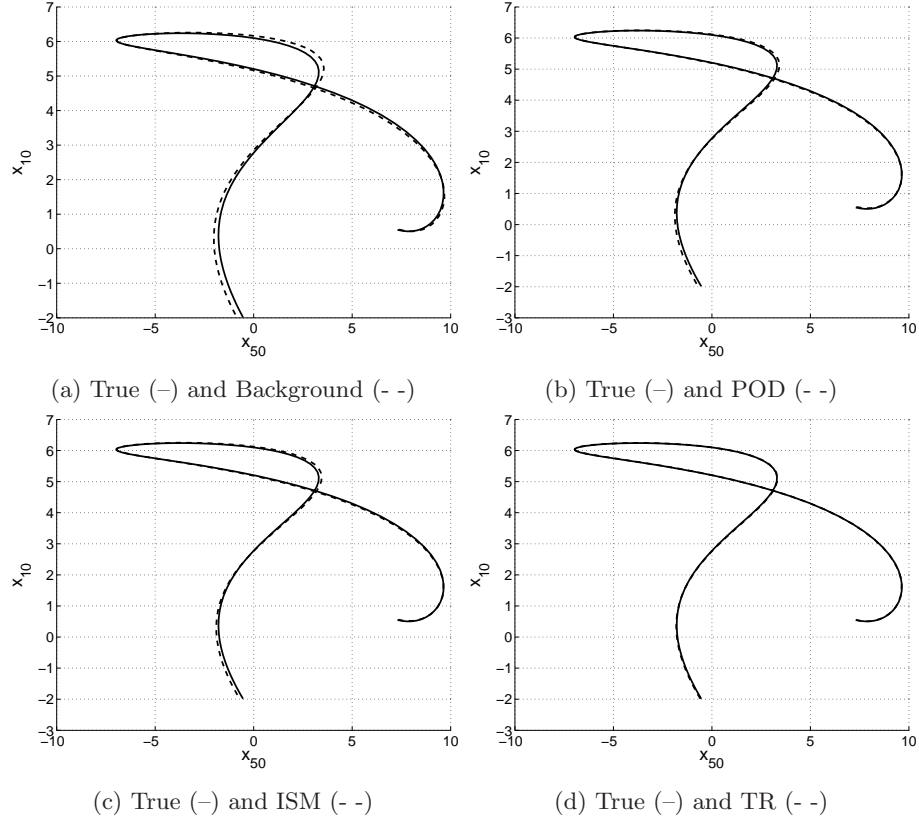


Figure 5: Snapshots of the 4D-EnKF implementations POD, SVD, ISM, and TR for the model resolution $n = 10^5$. The x_{50} and x_{10} components are plot for each method (- -) and the true solution (-).

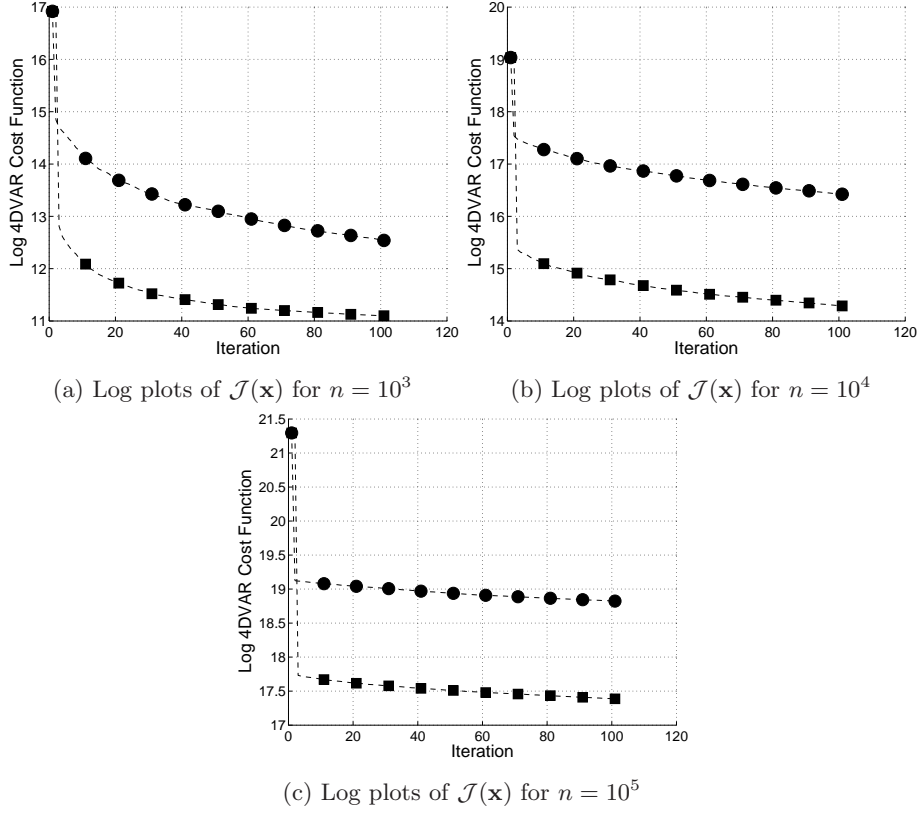


Figure 6: Cost function values for each iteration of the iterative 4D-EnKF implementations ISM ($-\bullet$) and TR ($-\blacksquare$) and for different Lorenz model resolutions (n).

4.2 The Quasi-geostrophic model

The Earth's ocean has a complex flow system influenced by the rotation of the Earth, the density stratification due to temperature and salinity, as well as other factors. The quasi-geostrophic (QG) model is a simple model which mimics the real behavior of the ocean. It is defined by the following partial differential equation [13]

$$\omega_t + r \cdot J(\omega, \psi) + \beta \cdot \psi_x - v \cdot \nabla^4 \psi = -\mu \cdot \nabla^2 \psi + \tau \cdot \sin\left(\frac{2 \cdot \pi \cdot y}{L_y}\right), \quad (49)$$

in $\Omega = [0, L_x] \times [0, L_y]$, where, x and y represents the horizontal and vertical space components, ω is the vorticity, ψ is the stream function, $J(\omega, \psi)$ is the Jacobian of two fields

$$J(\omega, \psi) = \omega_x \cdot \psi_y + \omega_y \cdot \psi_x \quad (50)$$

and ∇^2 is the Laplacian operator. The coefficients β , ν , μ , and τ are associated with the horizontal vorticity, the horizontal friction, the biharmonic horizontal friction and the horizontal wind stress at the surface of the ocean, respectively. Moreover, the vorticity is related to the stream function by the equation

$$\omega = \nabla^2 \psi. \quad (51)$$

The spatial domain is $\Omega = [0, 1] \times [0, 1]$ and its interior is covered by a 64×64 computational grid. Homogeneous Dirichlet boundary conditions are used, $\psi = \omega = 0$ on $\partial\Omega$. The state vector has a dimension $n = 3969$ (it does not include the boundary points on the grid). Four scenarios are considered by varying the number of ensemble members and observations. The number of observations is $m = p \cdot n$ where p is the percentage of observed components from the model size n . The scenarios are summarized in the Table 2. For all problem instances the model coefficient values are $\beta = 2 \times 10^{-3}$, $\nu = 3 \times 10^{-3}$, $\tau = 2 \cdot \pi$, $r = 2 \times 10^{-2}$, and $\mu = 1 \times 10^{-3}$.

$N + 1$	p	m	n
40	0.5	1984	3969
	0.7	2778	
80	0.5	1984	
	0.7	2778	

Table 2: QG-model scenarios based on the number of ensemble members N and observations m .

Details regarding the numerical implementation of the QG model are given next.

- The partial derivatives are discretized using central finite differences.
- The Arakawa method [20] is utilized in order to compute the Jacobian (50).
- The numerical integration of the model (49) is performed making use of a fourth order Runge-Kutta method. Thus, model errors have the bound $\mathcal{O}(h^4)$ where h is the time step size.
- The computation of the Laplace operator is accelerated making use of Fourier transforms.
- The time step size is 10^{-3} which represents one hour in the ocean. The integration is performed for 100 hours.

Other parameters of the numerical simulation are described below.

- We consider initial vorticities $\omega_0 \in \mathbb{R}^{D_1 \times D_2}$ whose grid components have the form

$$\begin{aligned} \omega_0^{ij} = \sin(2 \cdot y_i \cdot x_j) &+ \cos(2 \cdot y_i \cdot x_j) \\ &+ \sin(2 \cdot y_i \cdot x_j) \cdot \cos(2 \cdot y_i \cdot x_j) \end{aligned} \quad (52)$$

for $1 \leq i \leq D_1$ and $1 \leq j \leq D_2$.

- The background states are built as follows

$$\boldsymbol{\omega}_0^b = (\mathbf{1}_n + \boldsymbol{\nu}) \cdot \boldsymbol{\omega}_0 \in \mathbb{R}^{n \times 1}, \quad \boldsymbol{\nu} \sim \mathcal{N}(\mathbf{0}_n, \mathbf{B}_0), \quad (53)$$

where \mathbf{B}_0 is given in (45). Note that, the initial vorticity $\boldsymbol{\omega}_0 \in \mathbb{R}^{D_1 \times D_2}$ is reshaped to a vector of dimension $n = D_1 \cdot D_2$.

- There are 100 snapshots taken of each ensemble member, one at each observation time.
- The data error covariance matrix \mathbf{R}_k varies among the 100 observation times. At time t_k the data error covariance matrix is given by

$$\mathbf{R}_k = \begin{cases} \mathbf{R}^1 = \sigma_O^{(1)2} \cdot \mathbf{I}, & k \in \{1, 5, 9, \dots\}, \\ \mathbf{R}^2 = \sigma_O^{(2)2} \cdot \mathbf{I}, & k \in \{2, 6, 10, \dots\}, \\ \mathbf{R}^3 = \sigma_O^{(3)2} \cdot \mathbf{I}, & k \in \{3, 7, 11, \dots\}, \\ \mathbf{R}^4 = \sigma_O^{(4)2} \cdot \mathbf{I}, & k \in \{4, 8, 12, \dots\}, \end{cases} \quad (54)$$

where $\sigma_O^{(1)} = 0.01$, $\sigma_O^{(2)} = 0.001$, $\sigma_O^{(3)} = 0.02$ and $\sigma_O^{(4)} = 0.005$. The distribution of the observational errors $\boldsymbol{\epsilon}_k \sim (\mathbf{0}_m, \mathbf{R}_k)$ is changed between observation times. This mimics, for instance, the use of different sets of sensors at different times in the ocean.

- Four linear observation operators are considered, one for each observation time. The error distribution of the measurements taken by each linear operator is associated with one data error covariance matrix from (54). The linear operators and their respective error distributions are shown in Figure 7. Thus, at time k , the observed components are given by

$$\mathbf{H}_k = \begin{cases} \mathbf{H}^1, & k \in \{1, 5, 9, \dots\}, \\ \mathbf{H}^2, & k \in \{2, 6, 10, \dots\}, \\ \mathbf{H}^3, & k \in \{3, 7, 11, \dots\}, \\ \mathbf{H}^4, & k \in \{4, 8, 12, \dots\}. \end{cases} \quad (55)$$

- The number of outer loops for the TR-4D-EnKF is equal to 100.
- The parameters for the TR-4D-EnKF optimization are $\gamma_{\text{inc}} = 1.01$, $\gamma_{\text{dec}} = 0.99$, $\Delta_{\text{max}} = 10^7$, $\Delta_0 = 7 \times 10^4$, $\eta = 0.1$, $\theta_1 = 0.25$ and $\theta_2 = 0.75$.

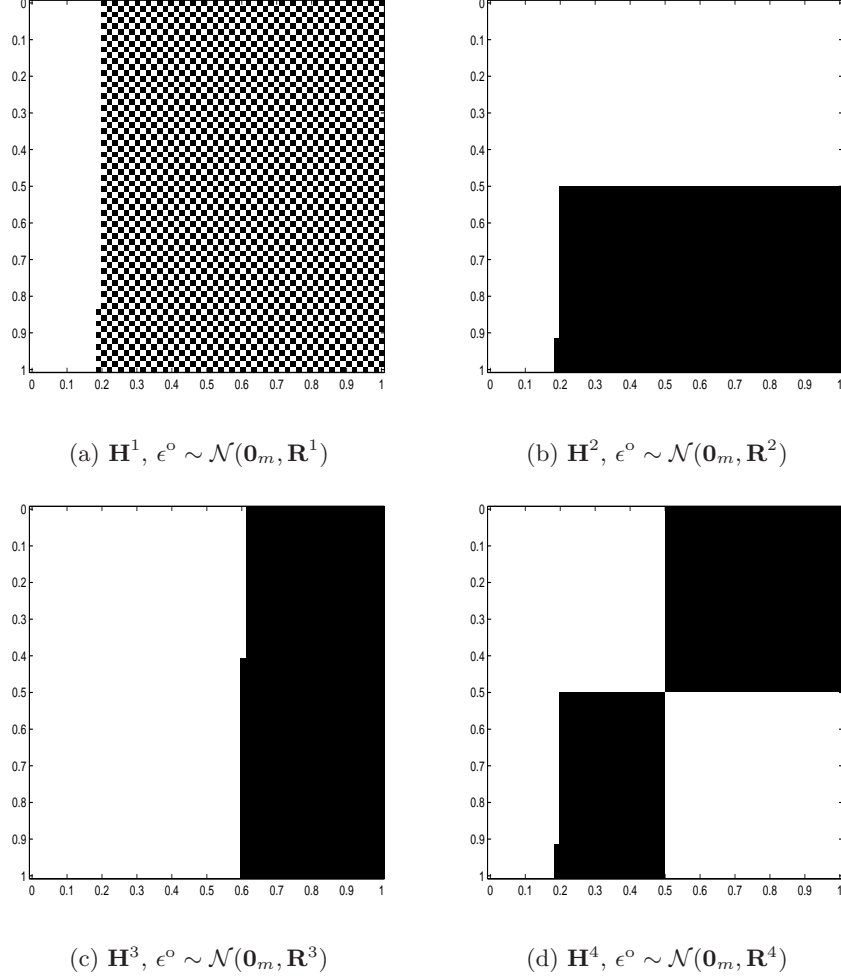


Figure 7: Linear observation operators and corresponding error distributions. The dark areas correspond to unobserved components.

Table 3 reports the cost function and RMSE values for the vorticity ω^a as well as the RMSE values of the stream function ψ^a for the different QG instances. The trajectories obtained by the TR-4D-EnKF after 100 outer iterations are compared with the trajectory of the initial background. Recall that the stream function can be recovered from the vorticity by the relation (51). The results shows the improvements obtained with the TR-4D-EnKF implementation. The results reported include the cost function values, the RMSE, and the computational time.

Method	$N + 1$	p	$\mathcal{J}(\omega)$	RMSE ω	RMSE ψ
Background	N/A	N/A	1.68×10^{18}	3.33×10^5	265.59
TR-4D-EnKF	40	0.5	1.91×10^{16}	5.70×10^4	23.49
		0.7	3.15×10^{16}	5.35×10^4	22.67
	80	0.5	1.72×10^{16}	5.47×10^4	20.92
		0.7	1.46×10^{16}	3.61×10^4	13.14

Table 3: Cost function values and RMSE for the TR-4D-EnKF implementation with different QG model instances.

The cost function values in the model space are decreased by two orders of magnitude from the background value. Figure 8 shows how the cost function value decreases with iterations. The initial cost function value corresponds to the background. Figure 9 shows how the approximation of the gradient in the ensemble space decreases with iterations. Recall that the projection of the model space onto the ensemble space is linear and therefore no curvature information is obtained. There is no guaranty that the optimal solution in the ensemble space is optimal in the model space. Nevertheless, the ratio ρ of the trust region framework, which relates the solution in the ensemble and model spaces, allows us to understand how well the current set of basis vectors performs in the optimization process.

After 100 TR-4D-EnKF outer iterations the improvements in the vorticity RMSE range from 82% to 89%. Likewise, the RMSEs of the stream functions are improved by 91% – 95%. The decrease of RMSE values for the stream function with iterations is shown in Figure 10. The background trajectory diverges quickly from the true state of the system due to the highly nonlinear behavior of the QG model. The TR-4D-EnKF analysis trajectory fits well the reference trajectory of the model throughout the assimilation window. This also can be seen in Figure 11 where solution snapshots are shown.

As expected, the ability of TR-4D-EnKF to ingest information from observations is related to the number of ensemble members. For instance, when the number of ensemble members is 40, increasing the number of observed components from 50% to 70% improves the TR-4D-EnKF analysis by $\sim 3\%$. On the other hand, when the size of the ensemble is 80, the same increase in the observations improves the solution by $\sim 34\%$. The more ensemble members are employed, the more information of the full system dynamics is captured by the ensemble, and the better the representation of model errors is. This leads to a better utilization of the information from observations.

On average, the CPU times per iteration of the TR-4D-EnKF method are 31.22 and 50.62 seconds for ensemble sizes of 40 and 80, respectively. This is a reasonable computational time given the considerable improvements obtained.

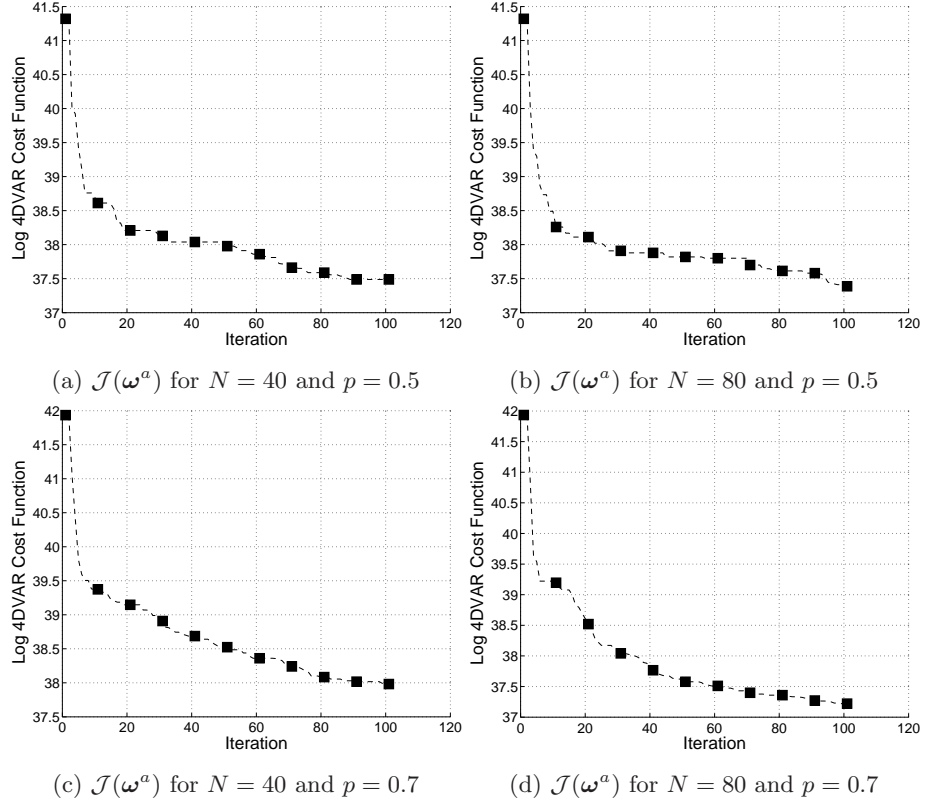


Figure 8: Cost function values in the model space among iterations for different test cases configurations.

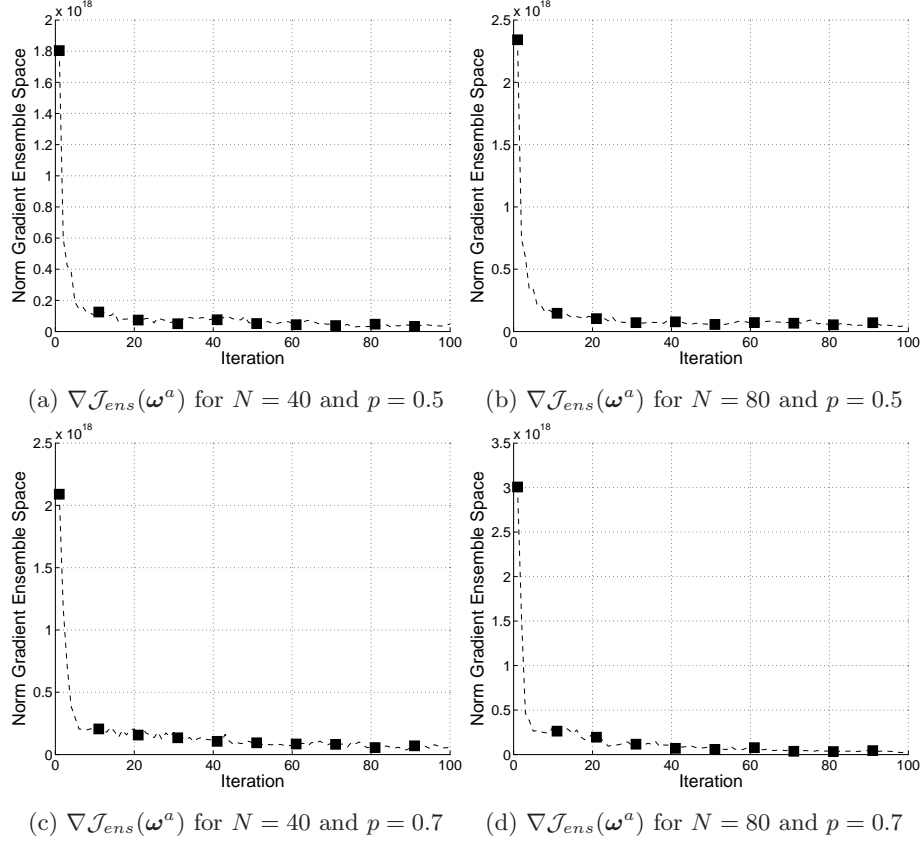


Figure 9: Gradient of the cost function in the ensemble space among iterations for different test cases configurations.

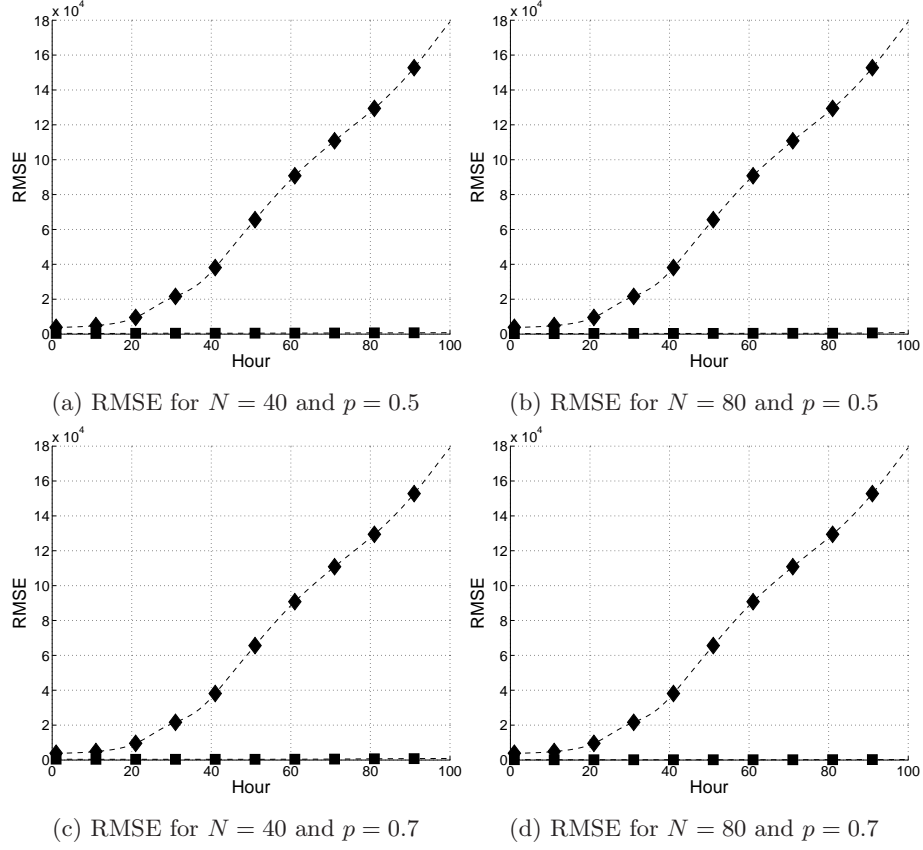


Figure 10: RMSE per hour for the initial background ($-\diamond$) and the TR-4D-EnKF solution ($-\blacksquare$) making use of different QG scenarios.

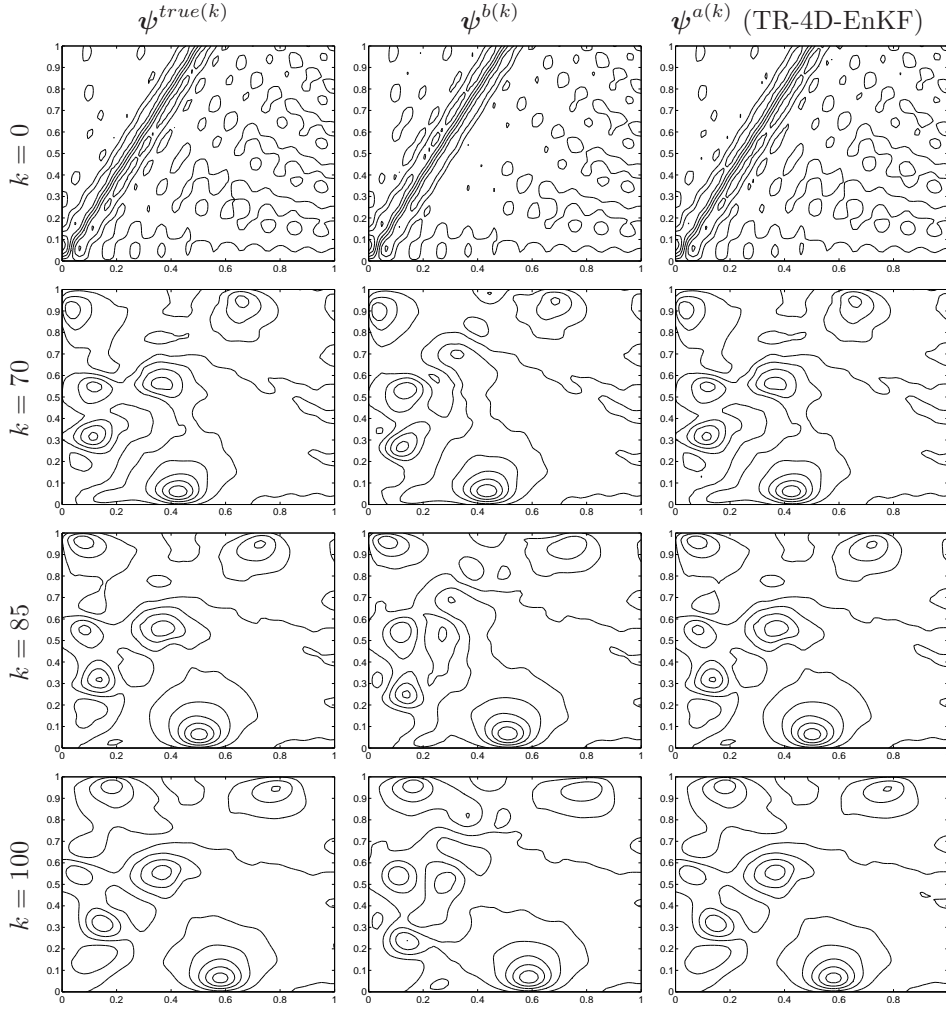


Figure 11: Snapshots of the true stream ψ^{true} , the initial background ψ^b and the analysis ψ^a obtained via 100 outer iterations of TR-4D-EnKF.

5 Conclusions

This paper develops TR-4D-EnKF, an ensemble-based 4D-Var data assimilation method based on the trust region framework. The proposed implementation projects the model space onto the space spanned by the deviations of the ensemble members from the mean, as is typically done in 4D-EnKF implementations. Small optimization problems are solved in the ensemble space. At each outer iteration a new ensemble based surrogate model of the 4D-Var cost function is constructed, and the convergence is controlled by the trust region method. The

trust region radius connects the optimal solution found in the ensemble space with the corresponding solution in the full model space. Moreover, the evolution of error statistics throughout iterations are captured by an empirical relation that uses the changes in trust region radius as a proxy for uncertainty decrease. Experimental results shows that the proposed implementation provide more accurate results than some of the best 4D-EnKF implementations available in the literature within a reasonable computational effort.

Acknowledgements

This work was supported in part by awards NSF CCF-1218454, AFOSR FA9550-12-1-0293-DEF, AFOSR 12-2640-06, and by the Computational Science Laboratory at Virginia Tech.

A The trust region method

Consider the unconstrained minimization problem

$$\mathbf{x}^* = \arg \min_{\mathbf{x}} f(\mathbf{x}) \in \mathbb{R}^{n \times 1}, \quad (56)$$

Trust Region based methods proceed as follows.

1. **Initialization.** Define the initial solution $\mathbf{x}^{[0]} \in \mathbb{R}^{n \times 1}$ and the parameters $\Delta_{\max} \in (0, \infty)$ (maximum radius size), $\Delta_{[0]} \in (0, \Delta_{\max})$ (initial radius size), $\eta \in (0, 2)$ (control variable for updating the solution), $0 < \theta_1 < \theta_2 < 1$ (control variables for updating the TR size), γ_{inc} (increasing factor of the TR size), γ_{dec} (decreasing factor of the TR size) and $j \leftarrow 0$ (iterate number).
2. **Model generation.** Build the quadratic model $m_{[j]}(\mathbf{s})$ as follows:

$$f(\mathbf{x}^{[j]} + \mathbf{s}) \approx m_{[j]}(\mathbf{s}) = f_{[j]} + \mathbf{g}_{[j]}^T \cdot \mathbf{s} + \frac{1}{2} \cdot \mathbf{s}^T \cdot \mathbf{G}_{[j]} \cdot \mathbf{s}. \quad (57)$$

In practice $f_{[j]} \approx f(\mathbf{x}^{[j]})$, $\mathbf{g}_{[j]} \approx \nabla f(\mathbf{x}^{[j]}) \in \mathbb{R}^{m \times 1}$, and $\mathbf{G}_{[j]} \approx \nabla^2 f(\mathbf{x}^{[j]}) \in \mathbb{R}^{m \times m}$, since the exact derivatives of (56) are unavailable or difficult to compute.

3. **Subproblem optimization.** Compute the optimal step via the solution of

$$\mathbf{s}^* = \arg \min_{\mathbf{s}} m_{[j]}(\mathbf{s}), \quad (58a)$$

$$\text{subject to } \|\mathbf{s}\| \leq \Delta_{[j]}. \quad (58b)$$

4. **Ratio computation.** Compute the ratio

$$\rho_{[j]} = \frac{f(\mathbf{x}^{[j]}) - f(\mathbf{x}^{[j]} + \mathbf{s}^*)}{m_{[j]}(\mathbf{0}_n) - m_{[j]}(\mathbf{s}^*)}, \quad (59)$$

where the numerator and denominator are often called the *actual* and *predicted* reduction.

5. **Solution update.** Update the solution $\mathbf{x}^{(k)}$ according to

$$\mathbf{x}^{[j+1]} = \begin{cases} \mathbf{x}^{[j]} & \text{for } \rho \leq \eta \\ \mathbf{x}^{[j]} + \delta \mathbf{x}^* & \text{otherwise} \end{cases}. \quad (60)$$

6. **Radius update.** Update the radius $\Delta_{[j]}$ according to

$$\Delta_{[j+1]} = \begin{cases} \Delta_{[j]} \cdot \gamma_{\text{dec}} & \text{for } \rho < \theta_1 \\ \Delta_{[j]} & \text{for } \theta_1 \leq \rho < \theta_2 \\ \Delta_{[j]} \cdot \gamma_{\text{inc}} & \text{for } \theta_2 \leq \rho \leq 1 \end{cases}. \quad (61)$$

7. **Iteration update.** Let $j \leftarrow j + 1$ and go to 2.

References

- [1] G. Candiani, C. Carnevale, G. Finzi, E. Pisoni, and M. Volta. A Comparison of Reanalysis Techniques: Applying Optimal Interpolation and Ensemble Kalman Filtering to Improve Air Quality Monitoring at Mesoscale. *Science of the Total Environment*, 458:7–14, AUG 1 2013.
- [2] X. Chen, I. M. Navon, and F. Fang. A Dual-weighted Trust-region Adaptive POD 4D-Var Applied to a Finite-element Shallow-water Equations Model. *International Journal for Numerical Methods in Fluids*, 65(5):520–541, 2011.
- [3] H. Cheng, Jardak M., Alexe M., and A. Sandu. A Hybrid Approach to Estimating Error Covariances in Variational Data Assimilation. *Tellus A*, 62(A):288–297, 2010.
- [4] K. S. Chung, W. Chang, L. Fillion, and M. Tanguay. Examination of Situation-Dependent Background Error Covariances at the Convective Scale in the Context of the Ensemble Kalman Filter. *Monthly Weather Review*, 141(10):3369–3387, OCT 2013.
- [5] A. R. Conn, I. M. Gould, and P. L. Toint. 6. *Global Convergence of the Basic Algorithm*, chapter 8, pages 115–168. MOS-SIAM Series on Optimization, 2000.
- [6] A. R. Conn, I. M. Gould, and P. L. Toint. 7. *The Trust-Region Subproblem*, chapter 9, pages 169–248. MOS-SIAM Series on Optimization, 2000.

- [7] A. R. Conn, I. M. Gould, and P. L. Toint. *8. Further Convergence Theory Issues*, chapter 10, pages 249–306. MOS-SIAM Series on Optimization, 2000.
- [8] A.R. Conn, K. Scheinberg, and L. N. Vicente. *Introduction to Derivative-Free Optimization*. MPS-SIAM Book Series on Optimization, Philadelphia, Pennsylvania, 2009.
- [9] A.R. Conn, K. Scheinberg, and L. N. Vicente. *The Lorenz Equations: Bifurcations, Chaos, and Strange Attractors*. Springer-Verlag, New York, New York, 2013.
- [10] A. L. Custodio, H. Rocha, and L.N. Vicente. Incorporating Minimum Frobenius Norm Models in Direct Search. *Computational Optimization and Applications*, 46(2):265–278, 2010.
- [11] A. L. Custodio and L. N. Vicente. Using Sampling and Simplex Derivatives in Pattern Search Methods. *SIAM J. on Optimization*, 18(2):537–555, May 2007.
- [12] J. David, S. Philip, and A. Walter. Reduced-order Modeling: New Approaches for Computational Physics. *Progress in Aerospace Sciences*, 40(12):51–117, 2004.
- [13] G. Evensen. Sequential Data Assimilation with a Nonlinear Quasi-geostrophic Model using Monte Carlo Methods to Forecast Error Statistics. *Journal of Geophysical Research: Oceans*, 99(C5):10143–10162, 1994.
- [14] G. Evensen. *Data Assimilation: The Ensemble Kalman Filter*, chapter 14, pages 210–237. Springer, 2009.
- [15] G. Evensen. The Ensemble Kalman Filter for Combined State and Parameter Estimation. *Control Systems, IEEE*, 29(3):83–104, June 2009.
- [16] S. Gratton, P. Laloyaux, and A. Sartenaer. Derivative-free Optimization for Large-scale Nonlinear Data Assimilation Problems. *Quarterly Journal of the Royal Meteorological Society*, pages n/a–n/a, 2013.
- [17] T. Hamill and C. Snyder. A Hybrid Ensemble Kalman Filter–3D Variational Analysis Scheme. *Monthly Weather Review*, 128(8):2905–2919, October 2000.
- [18] V. Hung and T. Hien. Modeling and Control of Physical Processes using Proper Orthogonal Decomposition. *Mathematical and Computer Modelling*, 33(13):223–236, 2001.
- [19] V. Hung and T. Hien. Proper Orthogonal Decomposition for Flow Calculations and Optimal Control in a Horizontal CVD Reactor. *Quarterly of Applied Mathematics*, 60(4):631–656, 2002.

- [20] A. Kacimi, T. Aliziane, and B. Khouider. The Arakawa Jacobian Method and a Fourth-order Essentially Nonoscillatory Scheme for the Beta-plane Barotropic Equations. *International Journal of Numerical Analysis & Modeling*, 10:571–587, 2013.
- [21] A. Lorenc. Modelling of error covariances by 4D-Var data assimilation. *Quarterly Journal of the Royal Meteorological Society*, 129(595):3167–3182, 2003.
- [22] E. Niño, A. Sandu, and J.L. Anderson. An Efficient Implementation of the Ensemble Kalman Filter Based on Iterative Sherman Morrison Formula. *Procedia Computer Science*, 9(0):1064 – 1072, 2012.
- [23] E. D. Nino-Ruiz, A. Sandu, and J. Anderson. An Efficient Implementation of the Ensemble Kalman Filter Based on an Iterative ShermanMorrison Formula. *Statistics and Computing*, pages 1–17, 2014.
- [24] P.D. Olivier. A Comparison Of Reduced Order Model Techniques. In *System Theory, 2009. SSST 2009. 41st Southeastern Symposium on*, pages 240–243, 2009.
- [25] A. Sandu and T.F. Chai. Chemical Data Assimilation—An Overview. *Atmosphere*, 2(3):426–463, 2011.
- [26] A. Sandu and Cheng H. A Subspace Approach to Data Assimilation and New Opportunities for Hybridization. *International Journal for Uncertainty Quantification*, submitted, 2013.
- [27] H. Song, I. Hoteit, B. Cornuelle, X. Luo, and A. Subramanian. An Adjoint-Based Adaptive Ensemble Kalman Filter. *Monthly Weather Review*, 141(10):3343–3359, October 2013.
- [28] R. Stefanescu, A. Sandu, and I.M. Navon. POD/DEIM Reduced-Order Strategies for Efficient Four Dimensional Variational Data Assimilation. Technical Report TR 3, Virginia Polytechnic Institute and State University, March 2014, also submitted to Journal of Computational Physics.
- [29] X. Tian, Z. Xie, and A. Dai. An Ensemble-based Explicit Four-dimensional Variational Assimilation Method. *Journal of Geophysical Research: Atmospheres*, 113(D21):n/a–n/a, 2008.
- [30] G. Triantafyllou, I. Hoteit, X. Luo, K. Tsiaras, and G. Petihakis. Assessing a Robust Ensemble-based Kalman Filter for Efficient Ecosystem Data Assimilation of the Cretan Sea. *Journal of Marine Systems*, 125(SI):90–100, SEP 2013.
- [31] N. Yussouf, E. Mansell, L. Wicker, D. Wheatley, and D. Stensrud. The Ensemble Kalman Filter Analyses and Forecasts of the 8 May 2003 Oklahoma City Tornadoic Supercell Storm Using Single- and Double-Moment Microphysics Schemes. *Monthly Weather Review*, 141(10):3388–3412, OCT 2013.

- [32] M. Zhang and F. Zhang. E4DVar: Coupling an Ensemble Kalman Filter with Four-Dimensional Variational Data Assimilation in a Limited-Area Weather Prediction Model. *Monthly Weather Review*, 140(2):587–600, 2011.

